

CONTROL DATA

CORPORATION

CONTROL DATA®
6000 COMPUTER SYSTEMS

COBOL INSTANT
6000 VERSION 3



CONTROL DATA

CORPORATION

CONTROL DATA[®]
6000 COMPUTER SYSTEMS

COBOL INSTANT
6000 VERSION 3

New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

[illegible]

Additional copies of this manual may be obtained from the nearest Control Data Corporation sales office.

Address comments concerning
this manual to:

CONTROL DATA CORPORATION
Software Documentation
215 MOFFETT PARK DRIVE
SUNNYVALE, CALIFORNIA 94086

© 1971

Control Data Corporation
Printed in the United States of America

CONTROL DATA 6400/6500/6600 COBOL

The COBOL language is designed to simplify the programming of business data processing operations; it produces easily modifiable source programs that result in shorter program development time and low program conversion costs. COBOL source and object programs run under the control of SCOPE Version 3.3.

COBOL for the CONTROL DATA[®] 6000 series is upwards compatible with ANSI COBOL and also with 3000 COBOL.

Special Features:

Mass Storage input and output including Indexed Sequential file processing

SORT verb sorts files within COBOL program

Automatic table search using index names and the SEARCH and SET statements

Report Writer to produce printed reports automatically, or report page may be produced by user with LINAGE clause and WRITE statement

Full arithmetic facility including:

18-digit operands

DIVIDE with REMAINDER

COMPUTE with exponentiation

CORRESPONDING option with ADD and SUBTRACT

Segmentation and overlay of object program

ENTER and CALL verbs, Common-Storage section provide communication with separately compiled COBOL programs and also with FORTRAN or COMPASS programs

COPY and INCLUDE provide access to COBOL source library

RERUN provides memory dumps with restart at specified checkpoints

Remote Interactive capability for remote terminal input/output

PROGRAM EFFICIENCY HINTS

To reduce key punching:

- Use abbreviations where permitted.

- Use PIC clause rather than SIZE, CLASS, USAGE clauses.

To increase compilation efficiency:

- Restrict data and paragraph names to 9 characters or less.

- Eliminate unnecessary paragraph names.

- Reduce forward references.

To increase execution efficiency:

- Use same size sending and receiving fields.

- Make table and item sizes a multiple of 10 characters.

- Reduce subscripting.

- Subscript with literals instead of variables.

- Use COMPUTATIONAL-1 items or index-names as subscripts.

- Use COMPUTATIONAL-1 items as arithmetic variables.

- Restrict arithmetic items to 9 digits or less.

- Use standard labels.

- Use SYNCHRONIZED RIGHT clause for data frequently referenced.

- Use SAME RECORD AREA to save moves; SAME AREA to save space.

COBOL NOTATION

[] Enclosed elements are optional.

{ } Only one element may be selected.

... Repeat preceding bracketed material as needed.

COBOL words have preassigned meanings and appear in capitals.

COBOL words not underlined may be omitted.

Terms in small letters are supplied by the programmer.

Punctuation and special characters are required where shown.

IDENTIFICATION DIVISION

IDENTIFICATION DIVISION.

PROGRAM-ID. program-name.
[AUTHOR. [comment-sentences.]]
[INSTALLATION. [comment-sentences.]]
[DATE-WRITTEN. [comment-sentences.]]
[DATE-COMPILED. [current-date supplied by compiler.]]
[SECURITY. [comment-sentences.]]
[REMARKS. [comment-sentences.]]

ENVIRONMENT DIVISION

ENVIRONMENT DIVISION. CONFIGURATION SECTION.

SOURCE-COMPUTER.

format 1:

COPY library-name $\left[\text{REPLACING} \begin{Bmatrix} \text{literal-1} \\ \text{word-1} \\ \text{identifier-1} \end{Bmatrix} \text{ BY} \begin{Bmatrix} \text{literal-2} \\ \text{word-2} \\ \text{identifier-2} \end{Bmatrix} \left[\begin{Bmatrix} \text{literal-3} \\ \text{word-3} \\ \text{identifier-3} \end{Bmatrix} \text{ BY} \begin{Bmatrix} \text{literal-4} \\ \text{word-4} \\ \text{identifier-4} \end{Bmatrix} \right] \dots \right]$

format 2:

$\begin{Bmatrix} 6400 \\ 6500 \\ 6600 \end{Bmatrix}$

OBJECT-COMPUTER.

format 1:

COPY library-name $\left[\text{REPLACING} \begin{Bmatrix} \text{literal-1} \\ \text{word-1} \\ \text{identifier-1} \end{Bmatrix} \text{ BY} \begin{Bmatrix} \text{literal-2} \\ \text{word-2} \\ \text{identifier-2} \end{Bmatrix} \left[\begin{Bmatrix} \text{literal-3} \\ \text{word-3} \\ \text{identifier-3} \end{Bmatrix} \text{ BY} \begin{Bmatrix} \text{literal-4} \\ \text{word-4} \\ \text{identifier-4} \end{Bmatrix} \right] \dots \right]$

format 2:

$\begin{Bmatrix} 6400 \\ 6500 \\ 6600 \end{Bmatrix}$ [SEGMENT-LIMIT IS priority-number].

SPECIAL-NAMES.

format 1:

COPY library-name $\left[\text{REPLACING} \left\{ \begin{array}{l} \text{literal-1} \\ \text{word-1} \\ \text{identifier-1} \end{array} \right\} \text{BY} \right.$
 $\left. \left\{ \begin{array}{l} \text{literal-2} \\ \text{word-2} \\ \text{identifier-2} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{literal-3} \\ \text{word-3} \\ \text{identifier-3} \end{array} \right\} \text{BY} \left\{ \begin{array}{l} \text{literal-4} \\ \text{word-4} \\ \text{identifier-4} \end{array} \right\} \right] \dots \right].$

format 2:

$\left[\text{SWITCH integer-1} \right.$
 $\left. \left\{ \begin{array}{l} \text{ON STATUS IS switch-status-name 1} \\ \text{[OFF STATUS IS switch-status-name-2]} \\ \text{OFF STATUS IS switch-status-name-2} \\ \text{[ON STATUS IS switch-name-1]} \end{array} \right\} \right]$

[non-numeric-literal IS mnemonic-name-1] ...

[implementor-name IS mnemonic-name-2] ...

[CURRENCY SIGN IS literal]

[DECIMAL-POINT IS COMMA]

[CONSOLE IS mnemonic-name-3]

[TERMINAL IS mnemonic-name-4].

INPUT-OUTPUT SECTION.

FILE-CONTROL.

format 1:

COPY library-name $\left[\text{REPLACING} \left\{ \begin{array}{l} \text{literal-1} \\ \text{word-1} \\ \text{identifier-1} \end{array} \right\} \text{BY} \right.$
 $\left. \left\{ \begin{array}{l} \text{literal-2} \\ \text{word-2} \\ \text{identifier-2} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{literal-3} \\ \text{word-3} \\ \text{identifier-3} \end{array} \right\} \text{BY} \left\{ \begin{array}{l} \text{literal-4} \\ \text{word-4} \\ \text{identifier-4} \end{array} \right\} \right] \dots \right].$

format 2:

SELECT [OPTIONAL] file-name-1 [RENAMING file-name-2]

ASSIGN TO implementor-name-1 [FOR MULTIPLE {REEL
UNIT}]

[RESERVE {NO
integer-1} ALTERNATE [{AREA }
{AREAS}]

[FILE-LIMIT IS literal-1]

[ORGANIZATION IS {INDEXED SEQUENTIAL}
STANDARD]

[ACCESS MODE IS {SEQUENTIAL}
RANDOM]

[PROCESSING MODE IS SEQUENTIAL]

[{ACTUAL
SYMBOLIC} KEY IS data-name-2]

[SELECT ...] .

I-O-CONTROL.

format 1:

COPY library-name [REPLACING {literal-1
word-1
identifier-1} BY
{literal-2
word-2
identifier-2} [{literal-3
word-3
identifier-3} BY {literal-4
word-4
identifier-4}] ...] .

format 2:

[RERUN [ON file-name-1] EVERY {END OF REEL
integer-1 RECORDS
condition-name
OF file-name-2]

[SAME [{SORT
RECORD}] AREA FOR file-name-3, file-name-4
[file-name-5] ...]

[MULTIPLE FILE TAPE CONTAINS file-name-6
[POSITION integer-2] [file-name-7
[POSITION integer-3] ...]

PICTURE DESCRIPTION CODES

Data Characters

- A Alphabetic character
- X Alphanumeric character
- 9 Numeric character

Operation Symbols

- S Signed
- V Assumed decimal point location
- P Assumed decimal point scaling position

Replacement Characters

- Z Leading zeros replaced by blanks
- * Leading zeros replaced by * (check protection symbol)

Insertion Characters

- \$ Dollar sign; floating when more than one (dollar sign may be replaced by currency sign defined in SPECIAL-NAMES)
- ,
- / Slash (instead of comma)
- .
- B Blank
- 0 Zero
- Minus sign when item is negative, blank when positive; floating when more than one
- + Plus sign when item is positive, minus when negative; floating when more than one
- CR Credit symbol when item is negative, blank when positive
- DB Debit symbol when item is negative, blank when positive

DATA SPECIFICATIONS

	File Section			Common and Working Storage Sections				Constant Section			
	01	group	elem	77	01	group	elem	77	01	group	elem
REDEFINES	I										
SIZE			R	R			R	R			R
USAGE											
CLASS				R				R			
OCCURS	I			I	I			I	I		
POINT LOCATION	J	I			J	I			J	I	
SIGNED	J	I			J	I			J	I	
JUSTIFIED	J	I			J	I			J	I	
SYNCHRONIZED	J	I			J	I			J	I	
PICTURE	J	I			J	I			J	I	
Editing Clauses	J	I			J	I		I	J	I	I
COPY											
VALUE	K	K	C					V			V
FILLER	I			I	I			I	I		

C Legal only in defining values for condition names

I Illegal

R Required if PICTURE is not used

blank Optional

V Required

J Legal only on elementary 01 items

K Documentary only

DATA DIVISION

DATA DIVISION.

[FILE SECTION.]

[COMMON-STORAGE SECTION.]

[WORKING-STORAGE SECTION.]

[CONSTANT SECTION.]

[REPORT SECTION.]

File Description Entry (File Section Only)

format 1:

$$\begin{aligned} & \text{FD file-name COPY library-name} \left[\text{REPLACING} \left\{ \begin{array}{l} \text{literal-1} \\ \text{word-1} \\ \text{identifier-1} \end{array} \right\} \text{BY} \right. \\ & \quad \left. \left\{ \begin{array}{l} \text{literal-2} \\ \text{word-2} \\ \text{identifier-2} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{literal-3} \\ \text{word-3} \\ \text{identifier-3} \end{array} \right\} \text{BY} \left\{ \begin{array}{l} \text{literal-4} \\ \text{word-4} \\ \text{identifier-4} \end{array} \right\} \dots \right] \right]. \end{aligned}$$

format 2:

FD file-name

$$\left[\text{RECORDING MODE IS} \left[\left\{ \begin{array}{l} \text{BINARY} \\ \text{DECIMAL} \end{array} \right\} \right] \left[\left\{ \begin{array}{l} \text{HIGH} \\ \text{LOW} \\ \text{HYPER} \end{array} \right\} \text{DENSITY} \right] \right]$$

[FILE CONTAINS ABOUT integer-1 RECORDS]

[BLOCK CONTAINS [integer-2 TO] integer-3 {RECORDS
CHARACTERS}]

[RECORD CONTAINS [integer-4 TO] integer-5 CHARACTERS

[DEPENDING ON {RECORD-MARK}
data-name-1]

LABEL {RECORDS ARE} {STANDARD
RECORD IS} {OMITTED
data-name-2}

If label records are STANDARD:

VALUE OF $\left\{ \begin{array}{l} \text{ID} \\ \text{IDENTIFICATION} \end{array} \right\}$ IS $\left\{ \begin{array}{l} \text{literal-1} \\ \text{data-name-3} \end{array} \right\}$
 $\left[\begin{array}{l} \text{DATE-WRITTEN} \text{ IS } \left\{ \begin{array}{l} \text{literal-2} \\ \text{data-name-4} \end{array} \right\} \end{array} \right]$
 $\left[\begin{array}{l} \text{EDITION-NUMBER} \text{ IS } \left\{ \begin{array}{l} \text{literal-3} \\ \text{data-name-5} \end{array} \right\} \end{array} \right]$
 $\left[\begin{array}{l} \text{REEL-NUMBER} \text{ IS } \left\{ \begin{array}{l} \text{literal-4} \\ \text{data-name-6} \end{array} \right\} \end{array} \right]$
 $\left[\begin{array}{l} \text{RETENTION-CYCLE} \text{ IS } \left\{ \begin{array}{l} \text{literal-5} \\ \text{data-name-7} \end{array} \right\} \end{array} \right]$

If label records are data-name-2:

$\left[\begin{array}{l} \text{VALUE OF ENDING-TAPE-LABEL-IDENTIFIER} \\ \text{IS } \left\{ \begin{array}{l} \text{literal-6} \\ \text{data-name-8} \end{array} \right\} \end{array} \right]$
 $\left[\begin{array}{l} \text{LINAGE} \text{ IS } \left\{ \begin{array}{l} \text{integer-6} \\ \text{identifier-1} \end{array} \right\} \text{ LINES} \end{array} \right]$
 $\left\{ \begin{array}{l} \text{DATA} \left\{ \begin{array}{l} \text{RECORDS ARE} \\ \text{RECORD IS} \end{array} \right\} \text{ data-name-9 [data-name-10] ... } \\ \left[\begin{array}{l} \text{REPORTS ARE} \\ \text{REPORT IS} \end{array} \right\} \text{ report-name-1 [report-name-2] ... } \end{array} \right\}$
[SEQUENCED ON data-name-11 [data-name-12] ...] .

Sort File Description Entry (File Section Only)

format 1:

SD file-name COPY library-name $\left[\begin{array}{l} \text{REPLACING} \left\{ \begin{array}{l} \text{literal-1} \\ \text{word-1} \\ \text{identifier-1} \end{array} \right\} \text{ BY} \\ \left\{ \begin{array}{l} \text{literal-2} \\ \text{word-2} \\ \text{identifier-2} \end{array} \right\} \left[\begin{array}{l} \left\{ \begin{array}{l} \text{literal-3} \\ \text{word-3} \\ \text{identifier-3} \end{array} \right\} \text{ BY} \left\{ \begin{array}{l} \text{literal-4} \\ \text{word-4} \\ \text{identifier-4} \end{array} \right\} \end{array} \right] \dots \end{array} \right]$.

format 2:

SD file-name

[FILE CONTAINS ABOUT integer-1 RECORDS]

[RECORD CONTAINS [integer-2 TO] integer-3 CHARACTERS]

$\left[\begin{array}{l} \text{DATA} \left\{ \begin{array}{l} \text{RECORDS ARE} \\ \text{RECORD IS} \end{array} \right\} \text{ data-name-1 [data-name-2] ... } \end{array} \right]$.

Record Description Entry (File, Common-Storage, Working-Storage and Constant Sections)

format 1:

level-number data-name-1 [REDEFINES data-name-2]
COPY data-name-3 [FROM LIBRARY].

format 2:

level-number { data-name-1 } [REDEFINES data-name-2]
FILLER

[CLASS IS { ALPHABETIC
NUMERIC
ALPHANUMERIC
AN }]

[SIZE IS integer-1 { CHARACTERS
DIGITS }]

[USAGE IS { COMP
COMPUTATIONAL
COMP-1
COMPUTATIONAL-1
COMP-2
COMPUTATIONAL-2
DISPLAY
INDEX }]

[OCCURS integer-1 [TO integer-2] TIMES
DEPENDING ON data-name-1]
{ ASCENDING } KEY IS data-name-4 [data-name-5] ...]
[INDEXED BY index-name-1 [index-name-2] ...]

[SIGNED]

[SIGN IS data-name-6]

[POINT LOCATION IS { LEFT
RIGHT } integer-5 PLACES]

[{ JUST
JUSTIFIED } RIGHT]

[{ SYNC } { LEFT }]
 [{ SYNCHRONIZED } { RIGHT }]

[VALUE IS literal-1]

[{ PIC } IS character-string]
 [{ PICTURE }]

[{ ZERO SUPPRESS
CHECK PROTECT
FLOAT DOLLAR SIGN
FLOAT CURRENCY SIGN } [LEAVING integer-6 PLACES]]

[{ BWZ
BLANK WHEN ZERO }] .

format 3:

66 data-name-1 RENAMES data-name-2 [THRU data-name-3] .

format 4:

88 condition-name { VALUE IS
VALUES ARE } literal-1 [THRU literal-2]
 [literal-3 [THRU literal-4] ...] .

Report Description Entry (Report Section only)

format 1:

RD report-name [WITH CODE mnemonic-name-1]

COPY library-name [REPLACING { literal-1
 word-1
 identifier-1 } BY
 { literal-2
 word-2
 identifier-2 } [{ literal-3
 word-3
 identifier-3 } BY { literal-4
 word-4
 identifier-4 }] ...] .

format 2:

RD report-name [WITH CODE mnemonic-name-1]

[{ CONTROL IS
CONTROLS ARE } { FINAL
FINAL data-name-1[data-name-2] ... }]

$\left[\text{PAGE } \left\{ \begin{array}{l} \text{LIMIT IS} \\ \text{LIMITS ARE} \end{array} \right\} \text{integer-1 } \left\{ \begin{array}{l} \text{LINE} \\ \text{LINES} \end{array} \right\} \right.$
 $\left[\text{HEADING integer-2} \right] \left[\text{FIRST DETAIL integer-3} \right]$
 $\left[\text{LAST DETAIL integer-4} \right] \left[\text{FOOTING integer-5} \right]$

Report Group Description Entry (Report Section only)

format 1:

01 [data-name-1] COPY data-name-2 [FROM LIBRARY]

REPLACING $\left\{ \begin{array}{l} \text{literal-1} \\ \text{word-1} \\ \text{identifier-1} \end{array} \right\} \text{BY } \left\{ \begin{array}{l} \text{literal-2} \\ \text{word-2} \\ \text{identifier-2} \end{array} \right\}$

$\left[\left\{ \begin{array}{l} \text{literal-3} \\ \text{word-3} \\ \text{identifier-3} \end{array} \right\} \text{BY } \left\{ \begin{array}{l} \text{literal-4} \\ \text{word-4} \\ \text{identifier-4} \end{array} \right\} \right] \dots$

format 2:

01 [data-name-1]

$\left[\text{[CLASS IS]} \left\{ \begin{array}{l} \text{ALPHABETIC} \\ \text{NUMERIC} \\ \text{ALPHANUMERIC} \\ \text{AN} \end{array} \right\} \right]$

$\left[\text{LINE NUMBER IS } \left\{ \begin{array}{l} \text{integer-1} \\ \text{PLUS integer-2} \\ \text{NEXT PAGE} \end{array} \right\} \right]$

$\left[\text{NEXT GROUP IS } \left\{ \begin{array}{l} \text{integer-3} \\ \text{PLUS integer-4} \\ \text{NEXT PAGE} \end{array} \right\} \right]$

$\left[\text{SIZE IS integer-5 } \left\{ \begin{array}{l} \text{CHARACTERS} \\ \text{DIGITS} \end{array} \right\} \right]$

$\left[\text{[USAGE IS] DISPLAY} \right]$

TYPE IS {
 REPORT HEADING
RH
PAGE HEADING
PH
OVERFLOW HEADING
OH
 { CONTROL HEADING } { data-name-3 }
 { CH } { FINAL }
DETAIL
DE
 { CONTROL FOOTING } { data-name-4 }
 { CF } { FINAL }
OVERFLOW FOOTING
OV
PAGE FOOTING
PF
REPORT FOOTING
RF
}

Report Element Description (Report Section only)

level number [data-name-1]

[CLASS IS {
 ALPHABETIC
NUMERIC
ALPHANUMERIC
AN
}]

[COLUMN NUMBER IS integer-1]

[{
 ZERO SUPPRESS
CHECK PROTECT
FLOAT DOLLAR SIGN
FLOAT CURRENCY SIGN
} [LEAVING integer-2 PLACES]

[{ BLANK WHEN ZERO }
 [BWZ]

[GROUP INDICATE]

[{ JUSTIFIED } RIGHT]
 [JUST]

[LINE NUMBER IS {
 integer-3
PLUS integer-4
NEXT PAGE
}]

[{ PIC
PICTURE } IS character-string]

[POINT LOCATION IS { LEFT
RIGHT } integer-5 PLACES]

[RESET ON { data-name-2 }
FINAL]

[SIGNED]

[SIGN IS data-name-3]

[SIZE IS integer-6 { CHARACTERS
DIGITS }]

{ SOURCE IS { SELECTED data-name-4 }
LINE-COUNTER
PAGE-COUNTER }
SUM data-name-5 { data-name-6 } ... [UPON data-name-7]
VALUE IS literal-1 }

[[USAGE IS] DISPLAY].

TYPE clause allowed if level 01

NEXT GROUP clause allowed if level 01

USAGE SPECIFICATIONS

Element	Upper Limit
data-name	30 characters, 5 levels of qualifications
elementary item/literal	255 characters/digits
PERFORM nesting	15 levels in separate overlays, no limit in main overlay
level numbers	01-49, 66, 77, 88, FD, RD, SD
OCCURS...DEPENDING ON	1 per record description
library copies	5 levels of nesting
ACCEPT items	80 characters; 40 characters from console
PICTURE clause	30 symbols
arithmetic operand	18 digits
GO TO statement	100 procedure names
ALTER statement	100 procedure names
DISPLAY items	no limit
ENTER parameters	no limit
Total files, I/O devices, and reports	53
Total procedure names	depends on field length
Total external references	depends on field length

VALID MOVE OPERATIONS

Rec. Field Source Field	Elem. Binary	Elem. Alpha	Elem. BCD Num.	Elem. AN	Elem. Edit Num.	Elem. Edit AN	Group AN
Elem. Binary	Num. Bin.	X	Conv. Num.	Conv.† AN	Conv. Edit	Conv.† AN- Edit	TD AN
Elem. Alpha	X	AN	TD AN	AN	X	AN- Edit	AN
Elem. BCD Num.	Conv. Bin.	TD AN	Num.	AN†	Edit	AN- Edit	AN†
Elem. AN	X	TD AN	Num.	AN	Edit	AN- Edit	AN
Elem. Edit Num.	X	TD AN	X	AN	X	AN- Edit	AN
Elem. Edit AN	X	TD AN	X	AN	X	AN- Edit	AN
Group AN	TD AN	TD AN	TD AN	AN	X	AN- Edit	AN
Group Binary & Mixed	TD AN	TD AN	TD AN	TD AN	X	TD AN- Edit	TD AN
Zero	Num. Bin.	X	Num.	AN	Edit	AN- Edit	AN
Literal & Fig. Cons. AN	X	TD AN	X	AN	X	AN- Edit	AN
Literal Num.	Conv. Bin.	X	Num.	AN†	Edit	AN- Edit	AN

† Valid only when source is integer; others PD.

Any move to a binary or mixed group is treated as an alphanumeric move; a precautionary diagnostic is issued.

A move to a figurative constant or literal is illegal.

X	Illegal
AN	Alphanumeric
AN-Edit	Alphanumeric edited
Conv.	Conversion prior to move
Edit	Numeric edited
Num.	Numeric
Num. Bin.	Numeric binary
TD	Trivial diagnostic issued

PROCEDURE DIVISION

PROCEDURE DIVISION.

DECLARATIVES.

Section-name SECTION. declarative-sentence.
Paragraph-name. sentence-1 [sentence-2] ...
END DECLARATIVES.

ACCEPT identifier-1 [FROM mnemonic-name-1]

ADD {identifier-1
literal-1} [{identifier-2
literal-2} ...] identifier-n [ROUNDED]

[ON SIZE ERROR imperative-statement]

ADD {identifier-1
literal-1} [{identifier-2
literal-2} ...] {GIVING
TO}

identifier-m [ROUNDED] [identifier-n [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

ADD {identifier-1
literal-1} [{identifier-2
literal-2} ...] TO {identifier-3
literal-3}

GIVING identifier-m [ROUNDED] [identifier-n [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

ADD {CORR
CORRESPONDING} identifier-1 TO identifier-2 [ROUNDED]

[identifier-3 [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

ALTER procedure-name-1 TO [PROCEED TO] procedure-name-2

[procedure-name-3 TO [PROCEED TO] procedure-name-4] ...

CLOSE file-name-1 [{UNIT
REEL}] [WITH {NO REWIND
LOCK}]

[file-name-2 [{UNIT
REEL}] [WITH {NO REWIND
LOCK}}]] ...

COMPUTE identifier-1 [ROUNDED] [identifier-2 [ROUNDED]] ...

$\left\{ \begin{array}{l} \text{FROM} \\ = \\ \text{EQUALS} \end{array} \right\} \left\{ \begin{array}{l} \text{literal} \\ \text{arithmetic-expression} \\ \text{identifier-3} \end{array} \right\}$

[ON SIZE ERROR imperative-statement]

$\left\{ \begin{array}{l} \text{COPY} \\ \text{INCLUDE} \end{array} \right\} \text{library-name} \left[\text{REPLACING} \left\{ \begin{array}{l} \text{literal-1} \\ \text{word-1} \\ \text{identifier-1} \end{array} \right\} \text{BY} \right. \\ \left. \left\{ \begin{array}{l} \text{literal-2} \\ \text{word-2} \\ \text{identifier-2} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{literal-3} \\ \text{word-3} \\ \text{identifier-3} \end{array} \right\} \text{BY} \left\{ \begin{array}{l} \text{literal-4} \\ \text{word-4} \\ \text{identifier-4} \end{array} \right\} \right] \dots \right]$

DISPLAY $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{identifier-2} \\ \text{literal-2} \end{array} \right\} \dots \right]$

[UPON mnemonic-name]

DIVIDE $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \text{INTO identifier-2 [ROUNDED]}$

[identifier-3 [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

DIVIDE $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \left\{ \begin{array}{l} \text{BY} \\ \text{INTO} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier-2} \\ \text{literal-2} \end{array} \right\} \text{GIVING identifier-3}$

[ROUNDED] [identifier-4 [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

DIVIDE $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \left\{ \begin{array}{l} \text{BY} \\ \text{INTO} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier-2} \\ \text{literal-2} \end{array} \right\}$

GIVING identifier-3 [ROUNDED]

REMAINDER identifier-4

[ON SIZE ERROR imperative-statement]

ENTER COBOL.

ENTER LINKAGE.

$\left\{ \begin{array}{l} \text{ENTER} \\ \text{CALL} \end{array} \right\} [\text{language-name}] \text{routine-name}$

[USING parameter-list] .

EXAMINE identifier-1

$\left\{ \begin{array}{l} \text{TALLYING} \left\{ \begin{array}{l} \text{ALL} \\ \text{LEADING} \\ \text{UNTIL FIRST} \end{array} \right\} \text{literal-1} [\text{REPLACING BY} \\ \text{literal-2}] \\ \text{REPLACING} \left\{ \begin{array}{l} \text{ALL} \\ \text{LEADING} \\ \text{[UNTIL] FIRST} \end{array} \right\} \text{literal-3 BY literal-4} \end{array} \right\}$

EXIT.

$\left\{ \begin{array}{l} \text{EXIT PROGRAM.} \\ \text{RETURN.} \end{array} \right\}$

GENERATE identifier-1

GO TO [procedure-name-1]

GO TO procedure-name-2 [procedure-name-3 ...]

DEPENDING ON identifier-1

IF conditional-expression [THEN] $\left\{ \begin{array}{l} \text{statement-1} \\ \text{NEXT SENTENCE} \end{array} \right\}$

$\left[\text{THEN} \right] \left\{ \begin{array}{l} \text{OTHERWISE} \\ \text{ELSE} \end{array} \right\} \left\{ \begin{array}{l} \text{statement-2} \\ \text{NEXT SENTENCE} \end{array} \right\}$

Conditional expressions include:

$\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \\ \text{formula-1} \end{array} \right\} \left\{ \begin{array}{l} \text{IS [NOT]} \left\{ \begin{array}{l} \text{GREATER THAN} \\ \text{GR} \\ > \\ \text{LESS THAN} \\ \text{LS} \\ < \\ \text{GREATER-EQUAL TO} \\ \text{GQ} \\ \text{LESS-EQUAL TO} \\ \text{LQ} \\ \text{EQUAL TO} \\ \text{EQ} \\ = \\ \text{IS UNEQUAL TO} \\ \text{EQUALS} \\ \text{EXCEEDS} \\ \text{IS NQ} \\ \text{IS NGR} \\ \text{IS NLS} \end{array} \right\} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier-2} \\ \text{literal-2} \\ \text{formula-2} \end{array} \right\}$

{identifier-3} IS [NOT] {POSITIVE
NEGATIVE
ZERO}

identifier-4 IS [NOT] {NUMERIC
ALPHABETIC}

[NOT] {condition-name
switch-status-name}

INITIATE {report-name-1 [report-name-2] ...}
ALL

MOVE { {CORR
CORRESPONDING} identifier-2 }
literal-1
identifier-1 } TO

identifier-3 [identifier-4] ...

MULTIPLY {identifier-1}
literal-1 } BY identifier-2 [ROUNDED]

[identifier-3 [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

MULTIPLY {identifier-1}
literal-1 } BY {identifier-2}
literal-2 }

GIVING identifier-3 [ROUNDED]

[identifier-4 [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

NOTE character-string.

OPEN { INPUT file-name-1 {REVERSED
WITH NO REWIND}
[file-name-2 {REVERSED
WITH NO REWIND}] ...
OUTPUT file-name-3[WITH NO REWIND]
[file-name-4[WITH NO REWIND]] ...
{INPUT-OUTPUT}
I-O } file-name-5[file-name-6] ...

PERFORM procedure-name-1[THRU procedure-name-2]

PERFORM procedure-name-1 [THRU procedure-name-2]

{ identifier-1 }
{ integer-1 } TIMES

PERFORM procedure-name-1 [THRU procedure-name-2]

UNTIL condition-1

PERFORM procedure-name-1 [THRU procedure-name-2]

VARYING { index-name-1 }
{ identifier-1 } FROM { literal-1
{ index-name-2 }
{ identifier-2 } } BY

{ literal-2 }
{ identifier-3 } UNTIL condition-2 [AFTER { index-name-3 }
{ identifier-4 }]

FROM { literal-3
{ index-name-4 }
{ identifier-5 } } BY { literal-4
{ identifier-6 } } UNTIL condition-3

[AFTER { index-name-5 }
{ identifier-7 }] FROM { literal-5
{ index-name-6 }
{ identifier-8 } } BY

{ literal-6 }
{ identifier-9 } UNTIL condition-4]]

READ file-name-1 RECORD [INTO identifier-1] AT END
imperative-statement

READ file-name-1 RECORD [INTO identifier-2] INVALID KEY
imperative-statement

RELEASE record-name-1 [FROM identifier-1]

RETURN file-name-1 RECORD [INTO identifier-1] AT END
imperative-statement

SEARCH identifier-1 [VARYING { index-name-1 }
{ identifier-2 }]

[AT END imperative-statement-1]

WHEN condition-1 { imperative-statement-2 }
{ NEXT SENTENCE }

[WHEN condition-2 { imperative-statement-3 }
{ NEXT SENTENCE }] ...

SEARCH ALL identifier-1 [AT END imperative-statement-1]

WHEN condition-1 {imperative-statement-2}
NEXT SENTENCE }

SEEK file-name-1 RECORD [WITH KEY CONVERSION]

SET {index-name-1}
 {identifier-1} [{index-2}
 {identifier-2} ...]

TO {index-name-3}
 {identifier-3}
 {literal-1}

SET index-name-1 [index-name-2] ...

{UP BY } {identifier-1}
{DOWN BY } {literal-1}

SORT file-name-1 ON {DESCENDING
 ASCENDING} KEY identifier-1 [identifier-2] ...

[ON {DESCENDING
 ASCENDING} KEY identifier-3 [identifier-4] ...]

{INPUT PROCEDURE IS section-name-1 [THRU section-name-2]}
{USING file-name-2 }

{OUTPUT PROCEDURE IS section-name-3 [THRU section-name-4]}
{GIVING file-name-3 }

STOP {literal}
 {RUN } .

SUBTRACT {identifier-1}
 {literal-1} [{identifier-2}
 {literal-2} ...] FROM identifier-m

[ROUNDED] [identifier-n [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

SUBTRACT {identifier-1}
 {literal-1} [{identifier-2}
 {literal-2} ...] FROM {identifier-m}
 {literal-m}

GIVING identifier-n [ROUNDED]

[identifier-o [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

SUBTRACT { CORR
CORRESPONDING } identifier-1 FROM
 identifier-2 [ROUNDED] [identifier-3 [ROUNDED] ...]
 [ON SIZE ERROR imperative-statement].

TERMINATE { report-name-1 [report-name-2] ... }
ALL

USE AFTER STANDARD ERROR PROCEDURE ON { file-name
INPUT
OUTPUT
INPUT-OUTPUT
I-O }

USE { BEFORE
AFTER } STANDARD [{ BEGINNING
ENDING }] [{ REEL
FILE }]

LABEL { PROCEDURE
PROCEDURES } ON { file-name
INPUT
OUTPUT
INPUT-OUTPUT
I-O }

USE BEFORE REPORTING identifier-1 [identifier-2] ...

USE FOR KEY CONVERSION ON { ALL
file-name-1 [file-name-2] ... }

WRITE record-name-1 [FROM identifier-1]

[{ BEFORE
AFTER } ADVANCING { identifier-2 LINES
 integer-1 LINES
 mnemonic-name-1 }]

[AT { END-OF-PAGE
EOP } imperative-statement]

WRITE record-name-2 [FROM identifier-2] INVALID KEY
 imperative-statement

COBOL CONTROL CARD

Ten parameters are used to select compilation options. All are optional and may be specified in any order.

```
{ COBOL.  
{ COBOL (p1,p2,p3,p4,p5,p6,p7,p8,p9,p10)} [comments]
```

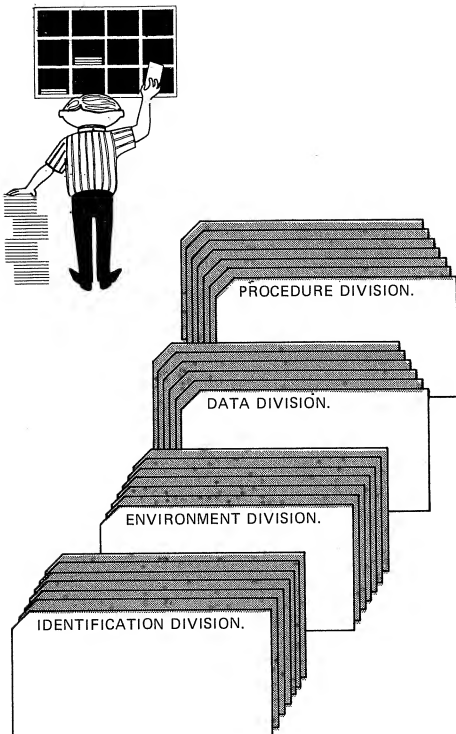
p1 (Source Input)	absent	}	INPUT assumed
	I		
	I = INPUT		
	I = fn		source input on file fn
p2 (Binary Output)	absent	}	relocatable binary file on file LGO
	B		
	B = LGO		
	B = fn		binary output on file fn
	B = 0		suppress binary output
p3 (List)	absent	}	normal listing on OUTPUT
	L		
	LX		extended diagnostics
	LR		cross reference pointers
	LC		copy from library
	LO		object code in octal
	LM		data map
	L = fn		output on file fn
	L = 0		suppress list output
p4 (Source Library)	absent	}	source library from file COLIB
	S		
	S = COLIB		
	S = fn		from file fn
p5 (Subcompile)	SUB		suppresses all data division binary output except from working and constant storage
p6 (Overlay Binary)	OB	}	binary output on LGO2
	OB = LGO2		
	OB = fn		binary output from overlay segments put on file fn

p7 (EDITLIB)	E = fn	add object code to system library using EDITLIB
p8 (ASCII Collating)	U	use ASCII collating sequence
p9 (Tape Sort)	T	sort requests tape sort
p10 (BCOMMON)	H	BCOMMON replaces blank common as buffer area

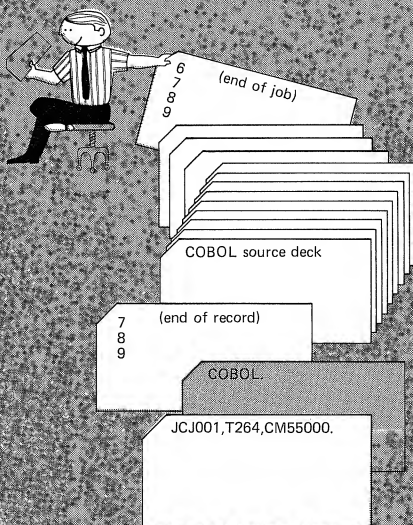
COBOL CODING FORMAT

<u>Column</u>	<u>Element</u>
1 - 6	Sequence number
7	Hyphen for continuation of words and/or literals
8	Division name Section name Paragraph name File description Record description level number
12	Record description data name First sentence of a paragraph File name Continuation of a data description or a sentence
73 - 80	Identification, optional
Sequence number	Optional, checked by the processor if used
Hyphen	Indicates continuation of a word from the preceding line
Division name	Terminated by period, remainder of line is blank
Section name	Followed by optional priority number, terminated by period, remainder is blank
Paragraph name	Terminated by period, and followed by at least one blank before text begins
File Description	FD or SD followed by file name and at least one blank
Record Description	Level number followed by at least one blank and data name
First Sentence	Begins in or after column 12. Spaces may be used freely to avoid splitting a word. If a word is split, a hyphen must appear in column 7 of the next line.

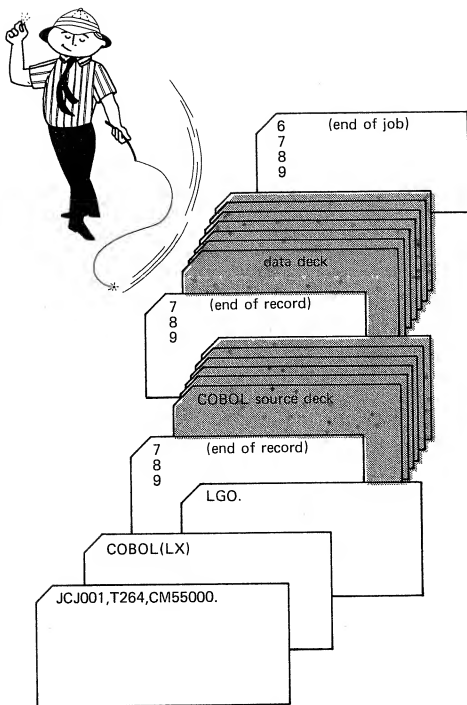
COBOL SOURCE DECKS



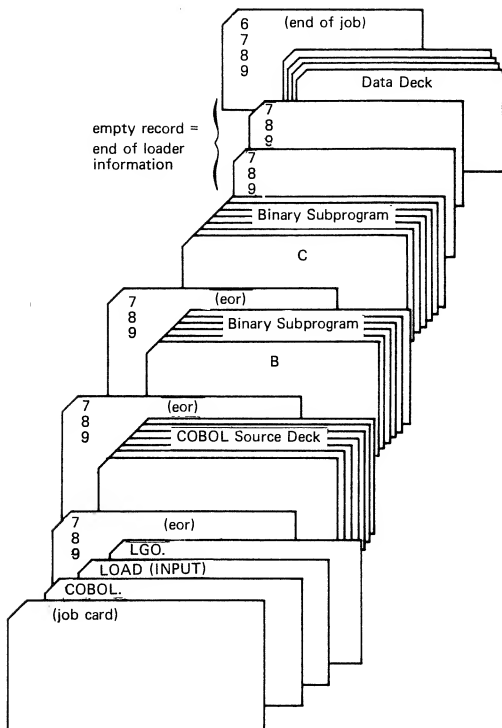
COBOL COMPILATION



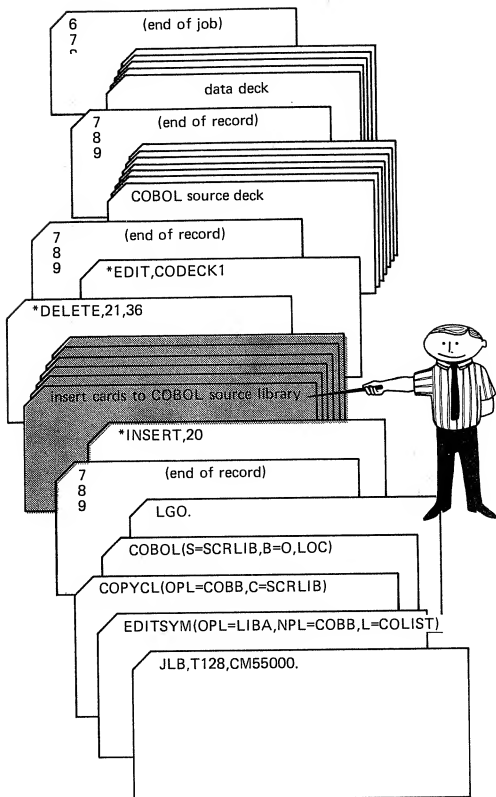
EXECUTION



EXECUTION WITH SEGMENTATION



LIBRARY UPDATE



COBOL RESERVED WORD LIST

*indicates word not implemented in 6000 COBOL.

ABOUT	CODE
ACCEPT	COLUMN
ACCESS	COMMA
ACTUAL	COMMON-STORAGE
ADD	COMP
*ADDRESS	COMP-1
ADVANCING	COMP-2
AFTER	COMPASS
ALL	COMPUTATIONAL
ALPHABETIC	COMPUTATIONAL-1
ALPHANUMERIC	COMPUTATIONAL-2
ALTER	COMPUTE
ALTERNATE	CONFIGURATION
AN	CONSOLE
AND	CONSTANT
*APPLY	CONTAINS
ARE	CONTROL
AREA	CONTROLS
AREAS	*CONVERSION
ASCENDING	COPY
ASSIGN	CORR
AT	CORRESPONDING
AUTHOR	CURRENCY
BEFORE	
BEGINNING	DATA
BEGINNING-FILE-LABEL	DATE-COMPILED
BEGINNING-TAPE-LABEL	DATE-WRITTEN
BINARY	DE
*BITS	DECIMAL
BLANK	DECIMAL-POINT
BLOCK	DECLARATIVES
BWZ	*DEFINE
BY	DENSITY
CALL	DEPENDING
CF	DESCENDING
CH	DETAIL
CHARACTER	DIGIT
CHARACTERS	DIGITS
CHECK	DISPLAY
CLASS	DIVIDE
*CLOCK-UNITS	DIVIDED
CLOSE	DIVISION
COBOL	DOLLAR
	*DOWN

EDITION-NUMBER	*HASHED
ELSE	HEADING
END	HIGH
END-OF-PAGE	HIGH-VALUE
ENDING	HIGH-VALUES
ENDING-FILE-LABEL	*HOLD
ENDING-TAPE-LABEL	HYPER
ENDING-TAPE-LABEL-IDENTIFIER	
ENTER	ID
ENTRY	IDENTIFICATION
ENVIRONMENT	IF
EOP	IN
EQ	INCLUDE
EQUAL	INDEX
EQUALS	INDEXED
ERROR	INDICATE
EVERY	INITIATE
EXAMINE	INPUT
EXCEEDS	INPUT-OUTPUT
EXIT	INSTALLATION
EXPONENTIATED	INTO
	INVALID
FD	I-O
FILE	I-O-CONTROL
FILE-CONTROL	IS
FILE-LABEL	
FILE-LIMIT	JUST
FILE-LIMITS	JUSTIFIED
FILLER	
FINAL	KEY
FIRST	*KEYS
FLOAT	
FOOTING	
FOR	LABEL
*FORMAT	LAST
FORTRAN-R	LEADING
FORTRAN-X	LEAVING
FROM	LEFT
	LESS
	LESS-EQUAL
GENERATE	LIBRARY
GIVING	LIMIT
GO	LIMITS
GQ	LINAGE
GR	LINAGE-COUNTER
GREATER	LINE
GREATER-EQUAL	LINE-COUNTER
GROUP	LINES

LINKAGE	PAGE
LOCATION	PAGE-COUNTER
LOCK	PERFORM
LOW	PF
LOW-VALUE	PH
LOW-VALUES	PIC
*LOWER-BOUND	PICTURE
*LOWER-BOUNDS	PLACES
LQ	PLUS
LS	POINT
	POSITION
*MEMORY	POSITIVE
MINUS	*PREPARED
MODE	PRIORITY
*MODULES	PROCEDURE
MOVE	PROCEDURES
MULTIPLE	PROCEED
MULTIPLIED	*PROCESS
MULTIPLY	PROCESSING
	PROGRAM
NEGATIVE	PROGRAM-ID
NEXT	PROTECT
NGR	PUNCH
NLS	PUNCHB
NO	
NOT	QUOTE
NOTE	QUOTES
NQ	
NUMBER	RANDOM
NUMERIC	RANGE
	RD
OBJECT-COMPUTER	READ
OCCURS	RECORD
OF	RECORD-MARK
OFF	RECORDING
OH	RECORDS
OMITTED	REDEFINES
ON	REEL
OPEN	REEL-NUMBER
OPTIONAL	RELEASE
OR	REMAINDER
ORGANIZATION	REMARKS
OTHERWISE	RENAMES
OUTPUT	RENAMING
OV	REPLACING
OVERFLOW	REPORT

REPORTING	SUPPRESS
REPORTS	SWITCH
RERUN	SYMBOLIC
RESERVE	SYNC
RESET	SYNCHRONIZED
RETENTION	
RETENTION-CYCLE	TALLY
RETURN	TALLYING
REVERSED	TAPE
REWIND	TAPE-LABEL
RF	TERMINAL
RH	TERMINATE
RIGHT	THAN
ROUNDED	THEN
RUN	THROUGH
	THRU
*SA	TIMES
SAME	TO
SD	TODAYS-DATE
SEARCH	TYPE
SECTION	
SECURITY	UNEQUAL
SEEK	UNIT
SEGMENT-LIMIT	UNTIL
SELECT	*UP
SELECTED	UPON
SENTENCE	*UPPER-BOUND
SEQUENCED	*UPPER-BOUNDS
SEQUENTIAL	USAGE
SET	USE
SIGN	USING
SIGNED	
SIZE	VALUE
SORT	VALUES
SOURCE	VARYING
SOURCE-COMPUTER	
SPACE	WHEN
SPACES	WITH
SPECIAL-NAMES	*WORDS
STANDARD	WORKING-STORAGE
STATUS	WRITE
STOP	
SUBTRACT	ZERO
SUM	ZEROES
*SUPERVISOR	ZEROS

COLLATING SEQUENCE

Collating Sequence	COBOL Character	Display Code	Hollerith Punch
00	Δ	55	space
01	≤*	74	8-5
02	[*	61	8-7
03	→*	65	0-8-5
04	≡*	60	0-8-6
05	∧*	67	0-8-7
06	↑*	70	11-8-5
07	↓*	71	11-8-6
08	>	73	11-8-7
09	≥*	75	12-8-5
10	└*	76	12-8-6
11	.	57	12-8-3
12)	52	12-8-4
13	;	77	12-8-7
14	+	45	12
15	\$	53	11-8-3
16	*	47	11-8-4
17	-	46	11
18	/	50	0-1
19	,	56	0-8-3
20	(51	0-8-4
21	=	54	8-3
22	≠†	64	8-4
23	<	72	12-0
24	A	01	12-1
25	B	02	12-2
26	C	03	12-3
27	D	04	12-4
28	E	05	12-5
29	F	06	12-6
30	G	07	12-7
31	H	10	12-8

*Not in COBOL character set; may be present in data

†COBOL quote character (") is output on printer as #

COLLATING SEQUENCE (Continued)

Collating Sequence	COBOL Character	Display Code	Hollerith Punch
32	I	11	12-9
33	V	66	11-0
34	J	12	11-1
35	K	13	11-2
36	L	14	11-3
37	M	15	11-4
38	N	16	11-5
39	O	17	11-6
40	P	20	11-7
41	Q	21	11-8
42	R	22	11-9
43	}††	62	0-8-2
44	S	23	0-2
45	T	24	0-3
46	U	25	0-4
47	V	26	0-5
48	W	27	0-6
49	X	30	0-7
50	Y	31	0-8
51	Z	32	0-9
52	:*	63	8-2
53	0	33	0
54	1	34	1
55	2	35	2
56	3	36	3
57	4	37	4
58	5	40	5
59	6	41	6
60	7	42	7
61	8	43	8
62	9	44	9

††COBOL record mark



CONTROL DATA
CORPORATION

**CORPORATE HEADQUARTERS, 8100 34th AVE. SO.
MINNEAPOLIS, MINN, 55440**

**SALES OFFICES AND SERVICE CENTERS
IN MAJOR CITIES THROUGHOUT THE WORLD**